

by Noah E. Crocker (former Robofest competitor)

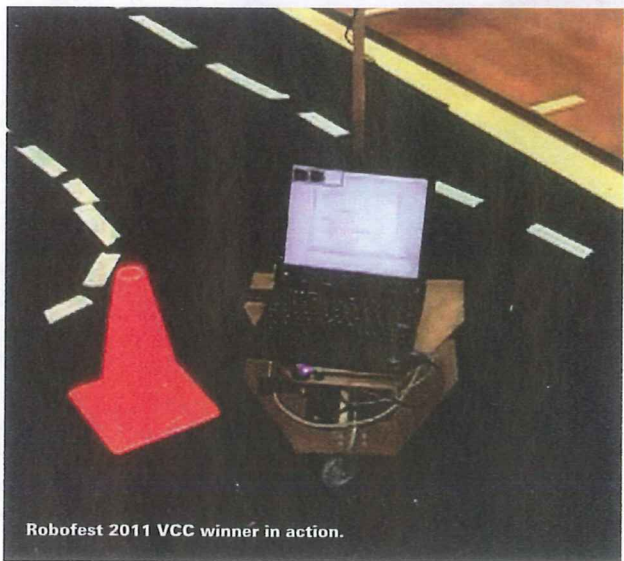
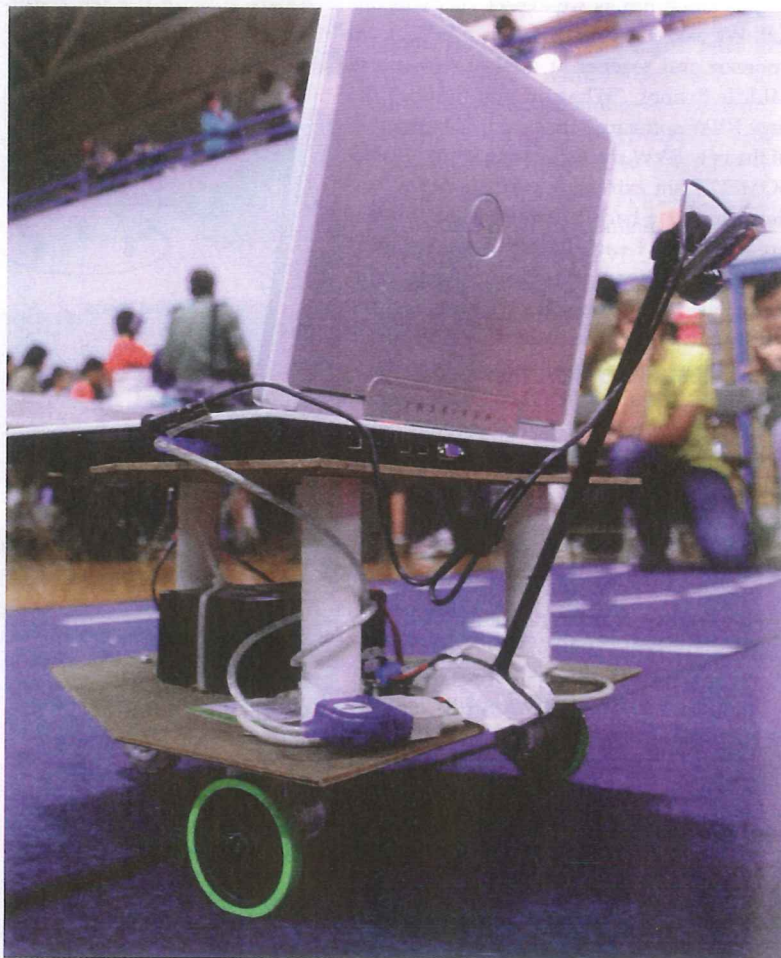
# Robofest 2011

## Vision Centric Challenge (VCC) Update!

In January of 2010, Miss Jamie MacLennan touched on Robofest's game and exhibition competition; however, Robofest's new Vision Centric Competition (VCC) was only briefly mentioned with the promise "more on this later." In an effort to deliver on that promise, I will provide a summary of this intriguing and challenging contest.

Each year, Robofest ([www.robofest.net](http://www.robofest.net)), sponsored by Lawrence Technological University in Southfield MI) hosts several robotics competitions. Among them are: "Game," a competition involving a mission for LEGO, VEX, or any other robots to complete in a set amount of time; "Exhibition," an open-ended competition in which students create a robot to solve practical problems or provide entertainment; and the VCC, the newest and most advanced of the three.

Unlike the Robofest divisions mentioned above which include middle school contestants, the Vision Centric Competition is reserved for high school and college students exclusively. The challenges are much more difficult to solve in comparison to the game and exhibition competitions, so some basic programming experience is very helpful. For this reason, Robofest hosts some workshops for new and aspiring programmers as well. Historically, this competition has required some pretty intense programming, so all hardware modifications are already



Robofest 2011 VCC winner in action.

completed. This unique aspect makes the VCC ideal for those who wish to become computer science or engineering majors, or simply want to dabble in some advanced programming requiring a high level of intuitive thinking without having to worry about mechanical design. The Vision Centric Competition is also a great driving force in developing an understanding of team oriented design environments set under strict time constraints.

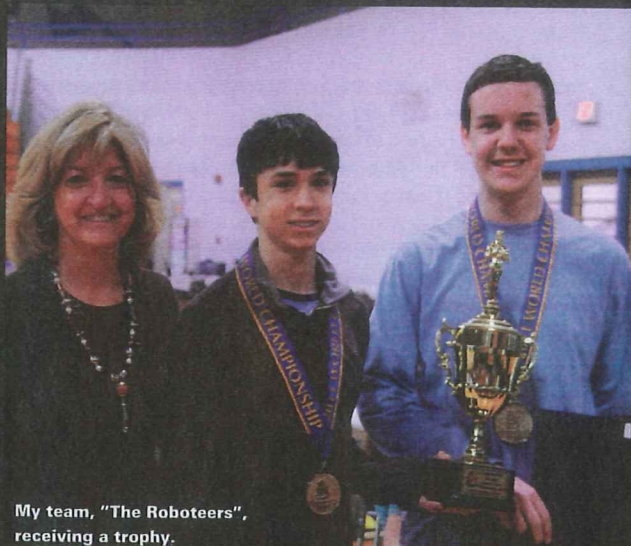
The VCC is the most exciting competition I have ever taken part in. While the Lego NXT (and RCX) Game competition was a great starting point for my interest in robotics, the VCC provided me with even more complicated challenges over the past two years. The Game competition offers a great place to start with robotics as it includes exciting multi-task challenges that combine mechanical design with simple algorithms. In contrast, VCC requires much more advanced problem solving and rewards complex programming. The ideas and creative spirits that are displayed in the Game competition are amazing, but the variety of ways that people approach VCC is truly incredible. No two teams had the same algorithm in VCC at any event.





## See the Video!

Scan barcode on your smartphone with a bar reader app or type in:  
[Find.botmag.com/111111](http://Find.botmag.com/111111)



My team, "The Roboteers", receiving a trophy.

## MY STORY

I graduated from high school this summer and will be studying Computer Science and Computer Engineering at the University of Michigan this fall. Ever since I was a kid I've loved electronics and computer programming. My parents suggested that I join FIRST robotics or Robofest. I ultimately decided to join Robofest for two reasons. FIRST robotics teams have a lot of members thus limiting individual contribution, whereas Robofest teams tend to be much smaller. FIRST robotics teams spend much of their time on mechanical design of a remote-controlled robot, whereas Robofest events are always completely autonomous. Given that I plan to study Computer Science, the Vision Centric Competition provided a richer challenge for me. Robofest has served as a natural outlet for my passion towards computers and autonomous electronic devices. I've had the pleasure of competing in two past Vision Centric Competition events. My experience in Robofest's game competitions gave me the basic foundation to take home first prize in both Vision Centric Competition events, and two \$2000 LTU scholarships.

It's those who think outside the box who excel, and if you're not one of those people, VCC is sure to help you with that. If you're looking for a good way to springboard into higher-tier competitions (like the Defense Advanced Research Projects Agency (DARPA) Urban Challenge, or the Mini Urban Challenges patterned after them, or the Intelligent Ground Vehicle Competition Navigation Challenge (IGVC), VCC is the best possible way to get you there.

### ROBOTICS PLATFORM

The Vision Centric Competition robotics platform has traditionally been the "Low cost Laptop roBOT", affectionately nicknamed "L2BOT"; although iRobot's "Create" has been permitted to compete as of last year. Most students opt for the L2BOT since it is a very versatile yet cheap platform. Some teams may not be able to handle the cost, so a few notable companies sometimes sponsor Vision Centric Competition teams. My team was sponsored by DENSO and U.S. Army Tank Automotive Research, Development and Engineering Center (TARDEC) with the agreement that we would return the L2Bot when the competition was over.

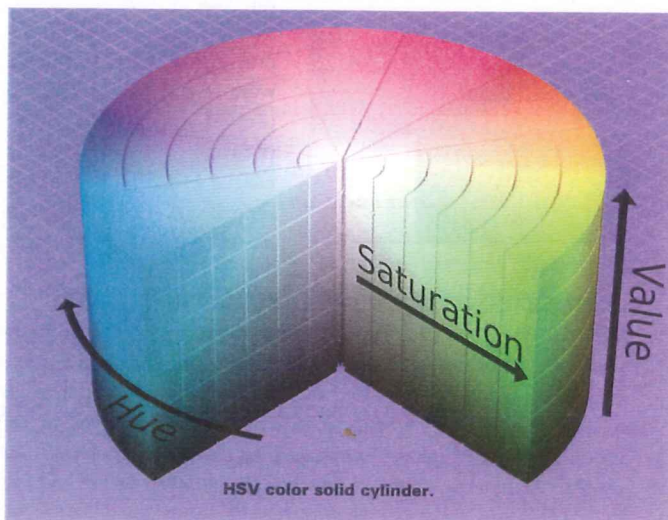
The L2BOT comes with a motor controller which was developed in-house by Brace Stout (a Lawrence Tech student) named "Low Cost Motor Controller" or LoCoMoCo. It has a serial interface which can be accessed using a USB to Serial converter. Unlike NXT motors, the LoCoMoCo has no rotation feedback capabilities, so you cannot directly measure how far you have traveled. L2BOTs are powered by



a single small 12V battery that can be charged using any car battery charger or a charger that can be purchased from Robofest. VCC is a camera-based competition; however, any additional sensors can be used. All processing is handled by an onboard laptop computer, typically running Java. Teams from China are using C++.

The first decision a team must make regarding image capture is resolution. Due to the processor intensive nature of video processing/segmentation, most teams use low camera resolutions. Typical resolutions fall in the range of 160x120 to as high as 640x480. Although the Java Media Framework (the tool used to capture and store images) can easily handle 1080p HD video, most laptops cannot perform all of the needed vision processing operations quickly enough to be ready for the next frame when it's available. In reactive robotics like VCC the number of decisions a robot can make in a single second is crucial. 160x120 normally offers plenty of accuracy.

The second decision a team must make is which color space to use. The three competing color spaces used with the L2BOT are RGB,





YUV, and HSV. RGB is the easiest for humans to conceptualize since any color can be made by mixing the 3 primary colors. YUV, on the other hand, is probably the hardest. YUV is beginning to overtake RGB as the most common color space, simply because it allows images to be compressed in higher quality than RGB with fewer processor cycles. The image capture code allows you to access either color space regardless of the input format. Unlike both RGB and YUV, the HSV color space is not natively supported by Java Media Framework. This means that a team who plans to use the HSV color space would need to write this conversion code themselves. This investment in writing conversion code, given the advantages of HSV, may well be worth it.

HSV (Hue, Saturation, Value) is easy to understand, and more versatile than the other color spaces. With RGB, a single pixel is described by how much red, green, and blue appear in that pixel. With HSV, a pixel is described by its hue (its color, as represented by an angle on the color wheel), saturation (colorfulness of the pixel relative to its own brightness), and value (brightness relative to the brightness of a similarly illuminated white). An example of how HSV is easier to use than RGB is efficiently filtering colors. If I wanted to filter all pixels that were above 50% red, then I would have to perform logic close to the following.

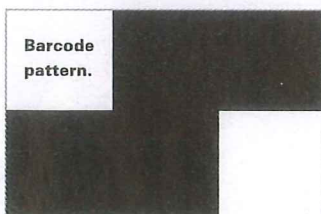
**RGB:** If the red component of the pixel is greater than the average of the blue and green components and the red component is greater than 50% then let the pixel pass through. The reason you have to check green and blue is best illustrated by the color white. White is a combination of 100% red, 100% green and 100% blue. If you only checked for Red and came across a white pixel, it would pass through the filter since red was above 50%. This is not the intended behavior.

**HSV:** If the Hue is close to Red, the saturation is above 50%, and the lightness is above 50%, then let the pixel pass through. In other words, if the pixel is red (Hue probably between about -20 and 20), and it's over 50% red (Saturation is high. The pixel is not gray, but colorful), and it's a bright shade of red (Value is high), then let it pass through.

## 2010 COMPETITION OVERVIEW

The 2010 challenge required teams to read a 2D barcode using the webcam. This barcode would inform the robot of the basic layout of the course. The bottom row would tell the robot how many obstacles were on the field, while the top row of the barcode would tell the robot if it should turn left or right at the first obstacle.

Top row, 011 = 3 cones. Bottom row, 110 = Pass on left side first. The robot would read the barcode, then navigate the course, passing each obstacle on the opposite side that it passed the last one, zig-zagging. Contestants were scored on a 2-tiered system. If your robot made it through the course, it was judged based on total elapsed time; if not, then it was judged based on how far it traveled.

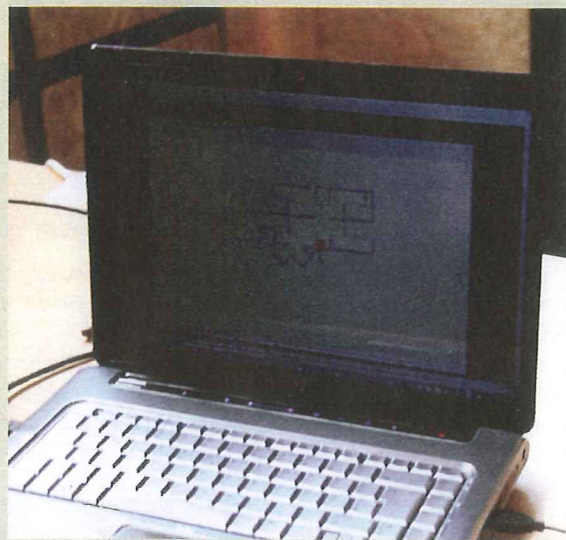


## CONNECTED COMPONENT LABELING ALGORITHM

Since 2010 was my first Vision Centric challenge, much of my prep time was spent doing research. My team read some of the papers that IGVC teams are required to publish. These papers detail the IGVC team's robot, algorithms, and strategies employed by their robot. Many IGVC teams were using an algorithm known as "connected component labeling" which is designed to detect and iden-

## Mapping with SLAM

After the 2011 challenge I thought I knew everything I needed to know to make a 100% reliable robot. I was wrong. After reading some more IGVC team papers, I learned about an algorithm called "SLAM". SLAM or "Simultaneous localization and mapping" is real time mapping algorithm. It does just what its name suggests. It finds the robot's position based on a map it created earlier, and simultaneously maps new objects. If no map has been made of the current area, it will just survey to create a map for later use. For example, the Roomba robot has recently been hacked to use SLAM to map its environment. The traditional Roomba navigates using reactive robotics principals (like most Vision Centric Competition teams) where it takes no past actions into account of future actions. Every few hours, it leaves its docking station, cleans the room, and returns to its docking station when low on batteries. It locates the docking station using an infra-red radar system. If the radar system is somehow hampered, the robot will have no idea



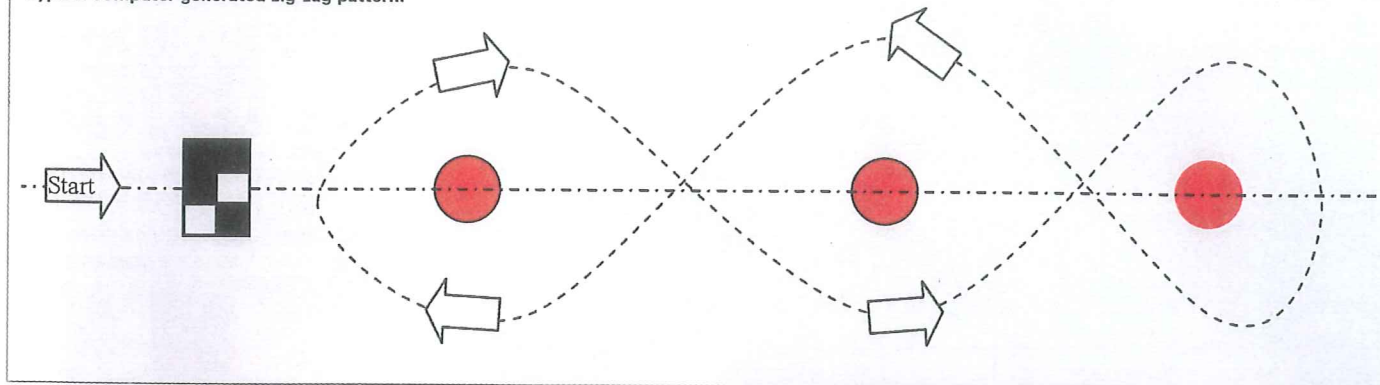
where to go and will sit in the middle of your room until you can tend to it to charge it up again. The hacked Roomba, however, is able to constantly know its location which results in better efficiency (it won't clean over what it has already cleaned) and eliminates the problem of getting lost. If the hacked Roomba needs to charge, it simply use the localization data provided by SLAM to figure out where it is, and then plot a path to get to the base station using the mapping data. This would be very useful in competitions like VCC (and it's used heavily in IGVC and DARPA), however, not many teams have tried it. Looking back now, SLAM would have been an invaluable addition to our arsenal of algorithms. Robotics is in its infant stages. There is always something new to learn, and always something unknown to discover. It's this aspect of robotics and Computer Science that I and many others love.

tify continuous masses within the camera image.

The programs that are given to teams at the start of the year all use image segmentation. It partitions an image based on some criteria and looks at each segment instead of every single pixel in the image. For example, a program that stops the robot when there are a lot of green pixels in the center of the screen is an example of image segmentation. Although image segmentation provides some basic function, Connected Component Labeling (CCL) is a much more reliable way to process an image since you can get the precise location of each visible object. CCL is used to detect groups



Typical computer generated zig-zag pattern.



of similar pixels and sort them into objects. The algorithm that my team decided to write makes two passes. The first pass populates an array with each pixel's object number. The second pass merges object numbers if the objects are adjacent but are somehow disconnected. After objects are detected and stored in an array, this information can be used by other algorithms. In our case, we used the pixel information to find the centroid (center) of the largest object. Think of the centroid as taking two averages. To find the centroid of an object in the x dimension, you divide the x position of every pixel in the object by the number of pixels in the object. The same is true for the y dimension as well.

By learning and coding this algorithm from scratch, my team was able to plot the quickest possible path through any course. This code was also very robust to course changes without requiring any "tuning". Since this algorithm is extremely useful, yet a little confusing, it will be added to the L2Bot code repository and handed out to this coming year's contestants. The material will also be explained in the L2Bot workshops.

### NEW CHALLENGES AT VCC 2011

Many new challenges were introduced in the 2011 challenge. The two-tiered judging criteria remained relatively unchanged, but the challenge itself took a few leaps in difficulty. Although on a much smaller scale, the 2011 Robofest Vision Centric Competition challenge was very much like the Intelligent Ground Vehicle Competition (IGVC: [www.igvc.org](http://www.igvc.org)) which requires very advanced algorithms and techniques like sensor fusion. In fact, IGVC and TARDEC invited participants of Robofest to visit IGVC for a day and hold a second competition with them. The challenge was to navigate



down a lane marked by dashed lines. This lane was full of obstacles, and the L2BOT was not allowed to touch any of them, nor leave the lane. After traveling to the end of the lane, the robot had to detect a "dead end" sign and navigate the course back to its start position.

My team reused the CCL code we wrote the year prior; however, we made a few modifications. We wrote a function that could bloat or shrink blobs (much like expanding a selection in Photoshop). We used the bloat function to expand blobs to half the width of the robot, meaning that pixels that were not part of any expanded blobs were "navigable", clear area. After researching several path finding algorithms (such as A\*, D\*flow, and Crowd Continuum), we decided to create our own path finding algorithm from scratch. The other algorithms were simply overkill given the parameters of the competition and would take far too many system resources.

Competing in VCC has been an excellent experience for me and my team. I've made new friends, learned a lot about programming and image segmentation/processing, learned how to work in a team environment, and experienced the thrill of competition. Vision Centric Competition actually caused me to shy away from pursuing a career as an airline pilot in favor of becoming a Computer Science major. Instead of piloting the airplanes you see in the sky every day, I'll be making the computers of the future that fly those very same planes. ☺

Links  
Robofest, [www.robofest.net](http://www.robofest.net)

For more information, please see our source guide on page 89

